GRID GENERATION FOR AEROSPACE APPLICATIONS

J. S. Mathur*

Abstract

Computational *Fluid* Dynamics is beginning to play a major *role in the design of aerospace vehicles. Recent* advances in numerical algorithms and *the development of high speed computers have made it possible to computeflows past practical configurations. Grid generation is the first step* in such a computation *andfor a geometrically complex three-dimensional configuration* this may *require significantly more time and effort than the flow analysis. Grid generation is therefore a major area of research and development, with the emphasis on creating tools which make the entire process faster.*

Various grid generation techniques have been developed for complex configurations. Multi-*block structured grids are the most popular, but the use of unstructured and Cartesian grids* is *rapidly* increasing. The grid *generation method adopted usually depends on the available flow solver code. No single method is superior in all respects, each method having its own advantages and disadvantages.*

Grid adaptation is *beginning to be recognised as an important procedure inflow computation. This allows grid points to be* concentrated in *those regions it-here higher* numerical *accuracy is required, since it* is computationally *expensive and often impractical to rise an excessively fine grid* in *the entire computational domain.*

This paper broadly discusses the various grid generation techniques available and their relative advantagesldisadvantages. It then discusses the grid generation methods used by the author and his colleagues for simulating the flow past various aerospace geometries like airfoils, wings, multi-body launch vehicles and aircraft configurations.

Introduction

The solution of problems in Computational Fluid Dynamics (CFD), using finite difference, finite volume or finite element techniques requires the discretisation of the computational domain into a set of points at which flow quantities are calculated. This set of points is called the computational grid. The accuracy of numerical computations is influenced to a large extent by the quality of the grid used, hence grid generation techniques should allow the user to control the grid structure and the distribution of grid points. Since grid generation is an intermediate stage in the flow simulation process, it should ideally be fast and automatic, requiring minimal user intervention. Despite tremendous advances in grid generation, we are still far away from this ideal 'fast and automatic' situation. For any complex three-dimensional geometry ---

1. Grid generation still takes a disproportionate amount of time and effort.
2. Some user expertise is usually required.
3. Simulations of flow over complex geometries are limited by the existing grid generation capabilities.

Hence, *there is a need* for software which enables fast and automatic grid generation and can be used by non-expert users. This has led to the development of a number 01 grid generation systems worldwide. Detailed descriptions of the many techniques which have been developed are available in [1], and a comprehensive overview of man! aspects of numerical grid generation is given in [2]. The most commonly used grids are the structured grids. Their inherent data structure allows the development of efficient *flow solvers, and* with the multi-block approach it is *possible to generate grids for extremely complex shape,*

Computational and Theoretical Fluid Dynamics Division, National Aerospace Laboratories, Post Box No. 1779
Bangalore-560 017, India

* *paper presented* during *the 2nd Annual CFD Symposium held on I I Aug 1999 at Bangalore*

The main drawback of this approach is the large amount of time and effort required for multi-block grid generation. Another disadvantage is the difficulty in implementing grid adaptation. Both algebraic and elliptic methods can be used for the generation of structured grids. Algebraic methods are extremely fast, but lack the inherent smoothing properties of the elliptic methods. The present trend is to use the fast algebraic methods for the initial grid followed, if needed, by the elliptic method for enhancing grid smoothness.

Unstructured grids are rapidly becoming popular, mainly because of the ability to generate such grids for complex shapes in a much shorter time frame. The data structures used are also suitable for grid adaptation, but require the development of specialised flow solvers.

Overlapped grids are also easy to generate, since only simple grid generation techniques are required for each component of any complex shape_ Such grids, although not widely used because of the specialised interpolation required in the flow solvers, can be used to solve some very complicated problems [3].

The easiest grids to generate are Cartesian grids. These Grids are not body conforming and require some special procedures for the accurate implementation of surface boundary conditions_ However, Cartesian grid generation is the fastest and the easiest to automate, and such a system is described in [4].

Grid adaptation is a technique to concentrate grid points in *regions of* high flow gradients and thus allow accurate flow simulations without the use of an excessively fine grid in the entire computational domain.There are two main techniques, point movement where grid points are moved from regions of low activity to *regions* of high activity, and point enrichment where additional grid points are added to regions of high activity.

The types of data structures traditionally used make it easier to implement point movement in structured grids, and point enrichment in unstructured grids. However, the use of more sophisticated data structures enables the point enrichment strategy to be used with structured grids.

Generalised two dimensional grids

Each type of grid has both advantages and disadvantages. This leads to the concept of hybrid grids which are a combination of different types of grids. The use of hybrid grids allows a variety of grid adaptation techniques io be used with both structured and Cartesian grids, along with unstructured grids. Hybrid grids require appropriate flow solvers, and in the finite volume framework, it is relatively easy to develop these solvers.

A two-dimensional finite volume flow solver has been developed which can work on hybrid grids consisting (jI a combination of triangularlquadrilateral cells. This flow solver, based on the kinetic flux vector splitting {KFVS) scheme for the Euler equations [5], has been applied to a variety of inviscid flow problems on different types of grids 16,7,$,9,101 and some illustrative results are shown here. Fig. I shows an adapted unstructured grid and computed contours of pressure for the transonic flow past a

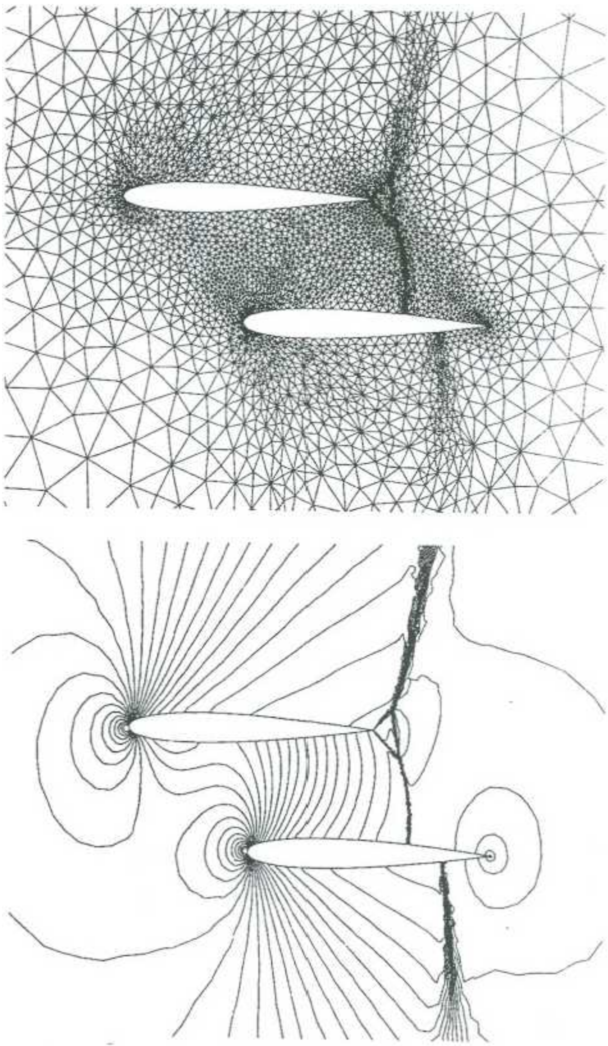


Fig. 1 Adapted unstructured grid and pressure contours NACA 0012 biplane, M=0.85, alpha = 0 Kinetic flux vector splitting method

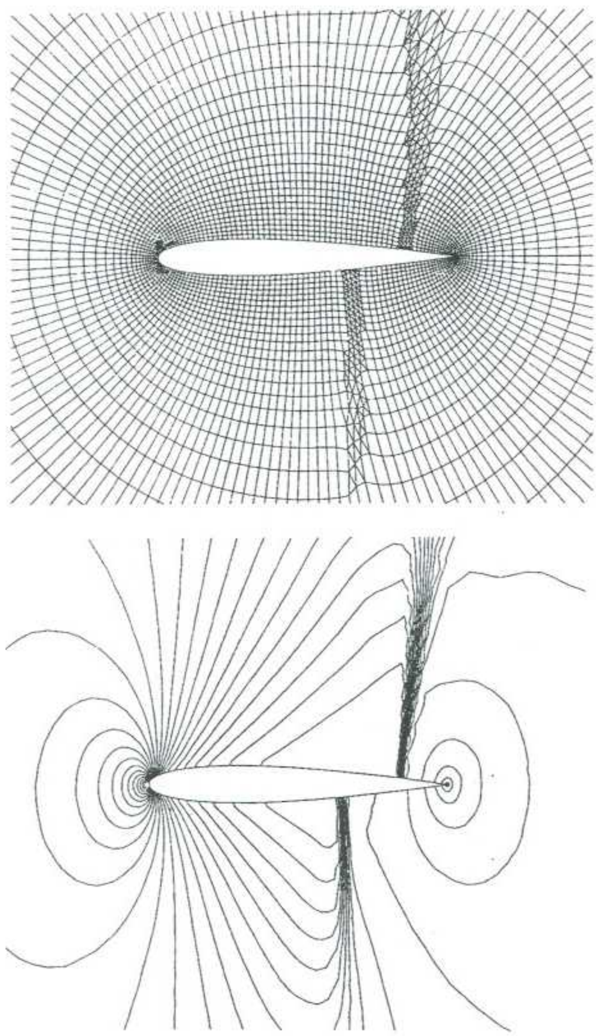


Fig. 2 Adapted structured (hybrid) grid and pressure contours NA CA 0012 airfoil, M = 0, 85, alpha = I Kinetic flux *vector* splitting *method*

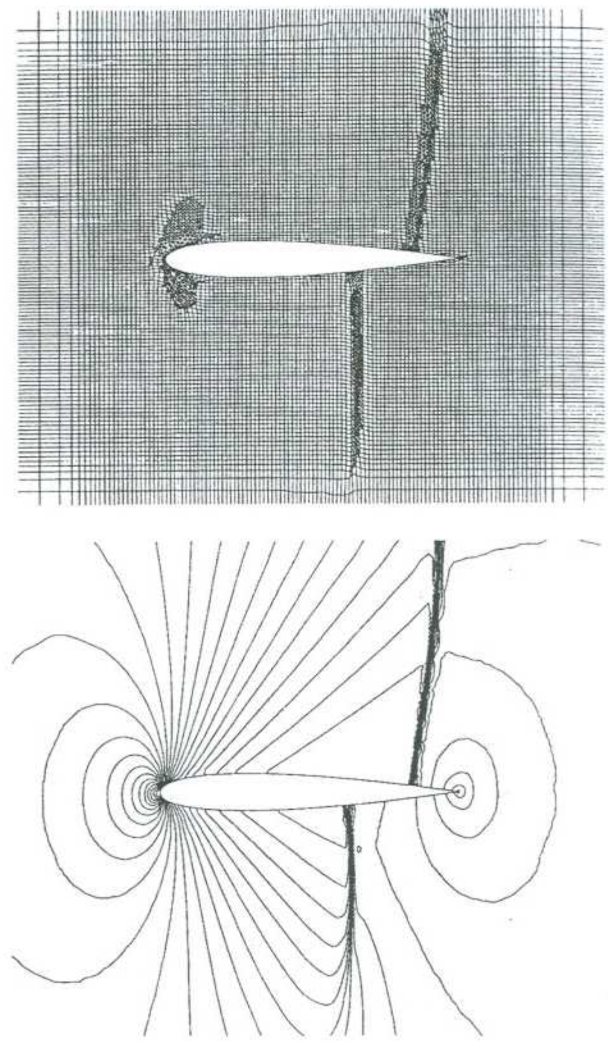


Fig. 3 Adapted Cartesian (body conforming, hybrid) grid and *pressure contours NA CA 0012 airfoil, M = 0.85, alpha = 1* Kinetic flux *vector* splitting *method*

NACA 0012 airfoil biplane configuration. The ability of the solution adaptive grid to capture crisply the triangular shock pattern near the trailing edge is clearly seen- Fig. 2 shows a hybrid adapted structured grid and computed contours of pressure for the transonic flow past a NACA 0012 airfoil. The original structured grid is adapted by the insertion of triangular grid cells, resulting in a sharper resolution of the shock waves. Fig.3 shows a similar hybrid adapted Cartesian grid and computed contours of pressure for the same flow case. Here, the Cartesian grid is first made body-conforming and then adapted by inserting triangular cells.

These results illustrate some of the types of general grids which can be used for flow computations using a generalised two-dimensional finite volume flow solver. The development of a similar three-dimensional solver would *result in* considerable flexibility in grid generation for complex geometries.

Three dimensional grids

All the three-dimensional flow solvers presently available in the National Aerospace Laboratories (NAL) have been developed for structured grids. Hence, grid generation activity for practical aerospace configurations is limited to structured grids.

Stacked grids

For relatively simple shapes like isolated wings, suitable three-dimensional grids can be generated by stacking sectional two-dimensional grids. These sectional grids can he generated by algebraic or -elliptic methods or a combination of the two. Figs. 4 and 5 show 0-11 and C-H type grids for the ONERA M6 wing, generated by this method. Such grids have been used for both inviscid [1I] and viscous [121, transonic flow computations. Fig. 6 shows a H-O type stacked grid for a cropped delta wing, which has been used to compute transonic viscous flow [131. The grid stacking method has also been extended, with care, to more complex geometries like launch vehicles [141. Fig. 7a shows a perspective view of the grid for a launch vehicle with two strap-on boosters. This is an extremely crude grid and merely illustrates the grid structure. Figs. 7h-7d show the actual sectional grids upstream of the vehicle, in the region of the core vehicle alone, and in the region of the core vehicle with the strap-on boosters, respectively. The computed results obtained on such grids are available in [14].

For aircraft configurations, grid stacking cannot give grids of the required quality. Hence, a full three-dimensional multi-block grid generation procedure is required.

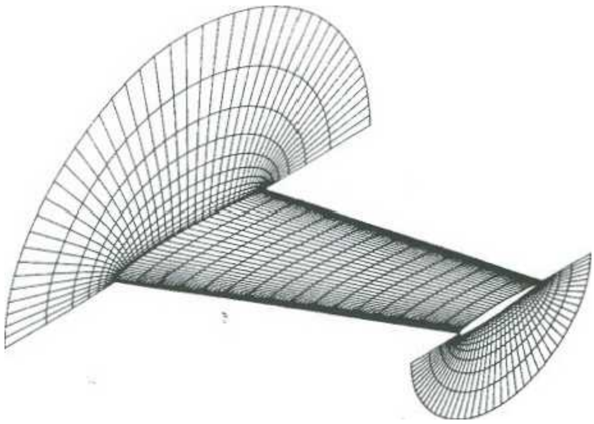


Fig. 4 Stacked 0-H type grid for ONERA M6 wing elliptic grid generation

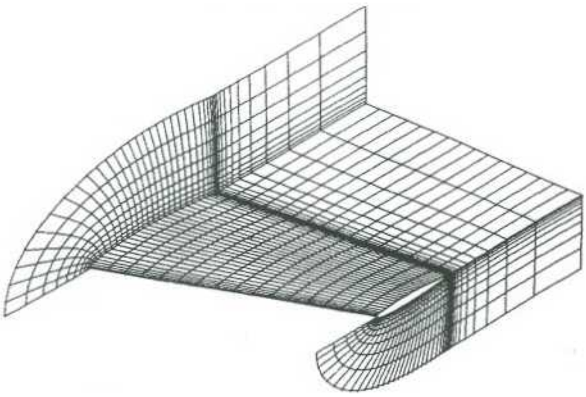


Fig. 5 Stacked C-H type grid for ONERA M6 wing algebraic grid generation

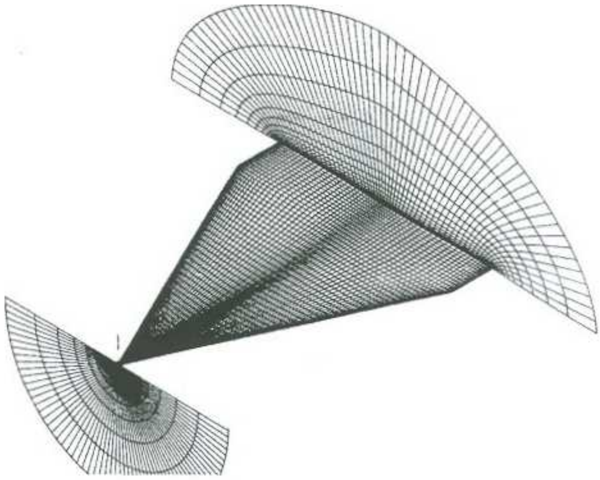


Fig. 6 Stacked H-O type grid for cropped delta wing elliptic/algebraic grid generation

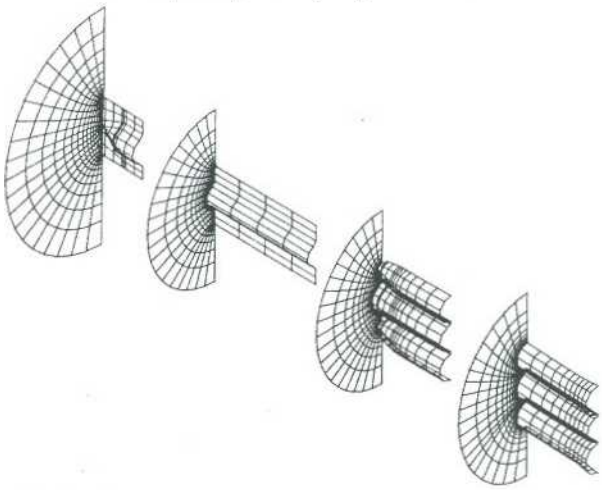


Fig. 7a Perspective view of stacked grid for launch vehicle with two strap-on boosters elliptic grid generation

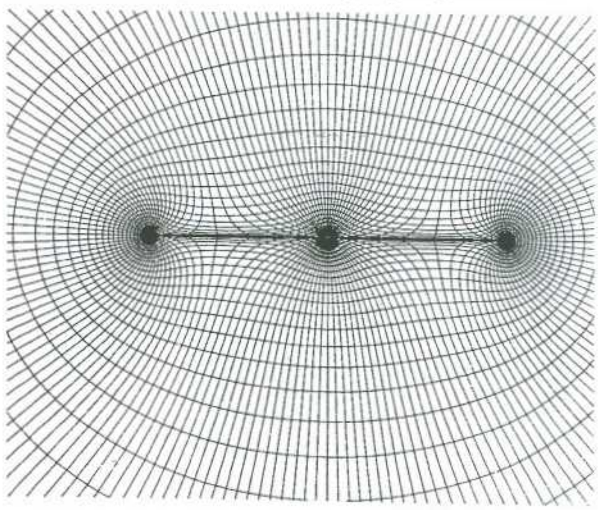


Fig. 7b Sectional grid for launch vehicle in region upstream of the vehicle

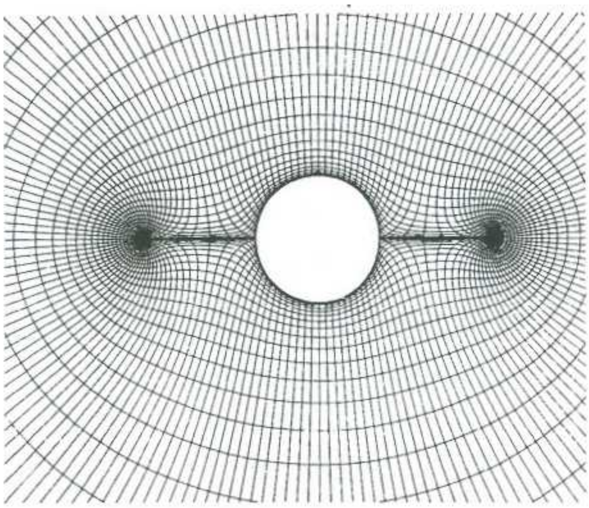


Fig. 7c Sectional grid for launch vehicle in region of core vehicle alone

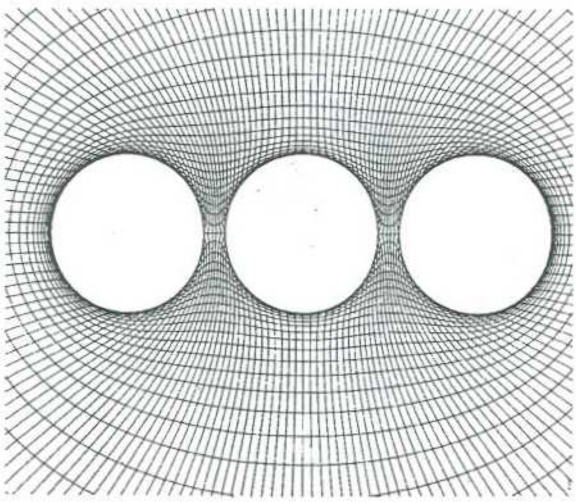


Fig. 7d Sectional grid for launch vehicle in region of care vehicle strap-on boosters

Multi-block grid generation

Mufti-block grid generation involves the following major steps

1. The user must first decide on the grid topology. This is based on the geometry, expected flow features and user experience. It is also based, to some extent, on the capabilities of the intended flow solver.
2. The chosen grid topology determines the nature of the surface grid required on the configuration. The geometry definition and surface grid generation is a ti me-consuming task.
3. The next step is the creation of the grids on all block faces, followed by the generation of the volume grid, taking care to maintain grid smoothness across block boundaries.
4. Finally, the block connectivity needs to be prescribed in the format required by the flow solver

Figure 8a shows the surface grid on the SARAS wing and fuselage for a 30 block multi-block grid generated using the JUMGRID code [15]. This code has been developed indigenously at NAL for the generation of multi-block grids around arbitrary three-dimensional bodies Fig.8h shows a view of the surface grid along with the field grid in the symmetry plane and a cross-sectional plane downstream of the fuselage. A close-up view of the surface and field grids near the wing-body junction is shown in Fig.8c. Finally, Fig.8d shows the computed Mach contours obtained on this grid using NAL's JUEL3D code [1 I]. Recently, NAL has also acquired the GRIDGEN software. This is a highly interactive and user friendly grid generation package, developed by M/s Pointwise Inc_, USA.

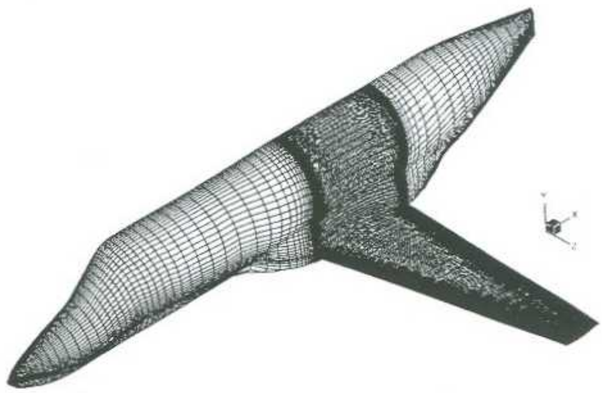


Fig. 8a Surface grid for SARAS wing fuselage 30 block multi-block grid

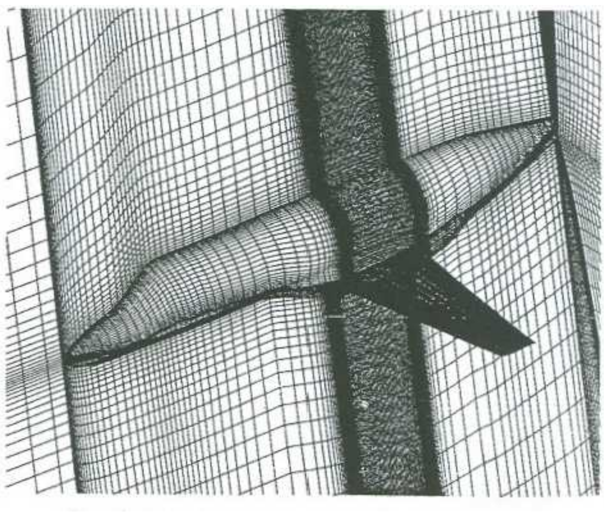


Fig. 8b View of surface and field grid for SARAS wing fuselage

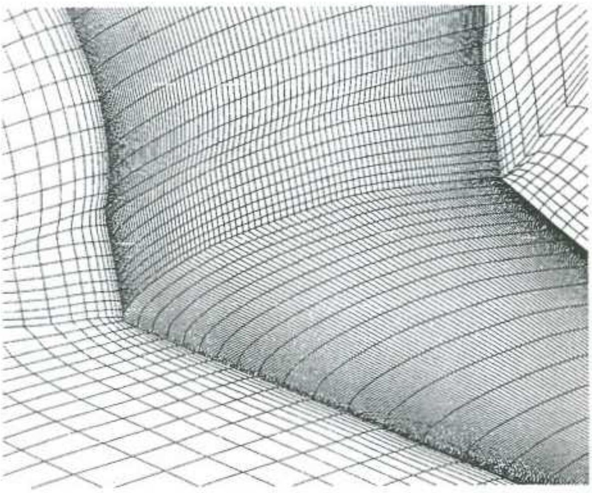
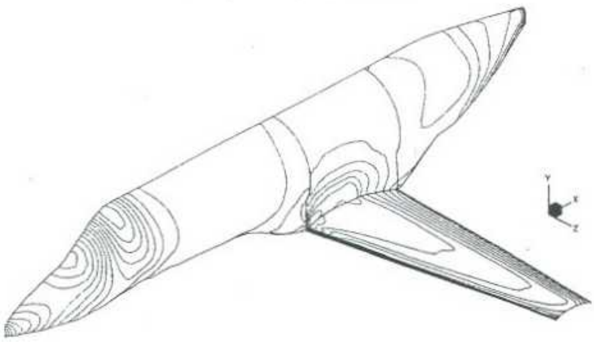


Fig. 8c Zoomed view of surface and field grid near wing fuselage junction



Fag. Sd Surface Mach contours for SARA. wing fuselage, M=0.5, alpha = 0

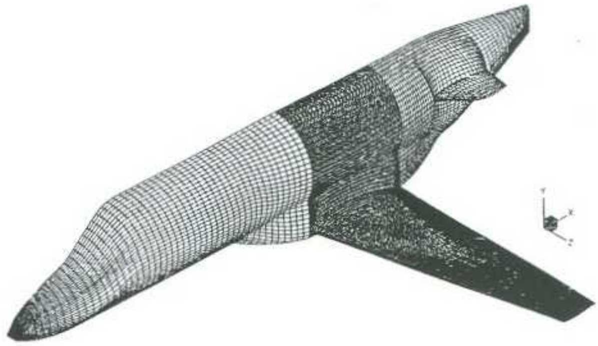


Fig. 9a Surface grid for SARAS wing fuselage stub-wing 64 block multi-block grid

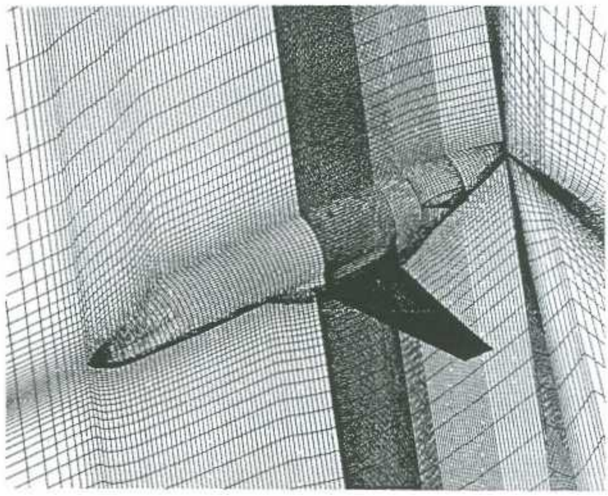


Fig. 9b View of surface and field grid for 5.ARAAS wine fuselage stub-Ywing

The GRIDGEN Software Package

GRIDGEN is a three-dimensional grid generation package. Earlier versions of GRIDGEN could generate only structured grids, but the latest version can generate unstructured grids as well. The main features of this package are

I. The package has a user friendly graphical interface which enables the grid to be visible at every stage. This is of immense help in generating complex multi-block grids with a large number of blocks.
2. Intersections of different components e.g. fuselage and wing are easily determined.
3. The user can control the number and distribution of grid points on all the block edges of the grid.
4. The initial grid is generated with transfmiie interpolation. Grid smoothing by elliptic equations, with control functions is possible.
5. A block can be checked for the presence of negative volumes and skewed cells. The grid axes can be independently set for each block and a warning is given if a left-handed system is prescribed. The blocks can he assembled in the desired order for the flow solver.

Figure 9a shows the surface grid on the SARAS wing, fuselage, and stub-wing for a 64 block grid generated using the GRIDGEN software package. Fig. 9b shows a view of the surface grid along with the field grid in the symmetry plane and a cross-sectional plane downstream of the fuselage. A close-up view of the surface and field grids near the wing-body junction is shown in Fig.9c. Finally, Fig.9d shows the computed Mach contours obtained on this grid using the JUEL3D code [I I].

Both the indigenous JUMGRID code and the commercial GRIDGEN code have similar capabilities in the generation of multi-block grids for complex geometries, The graphical interface makes the GRIDGEN code easier to use, particularly for grid smoothing, since the grid is continuously visible. With the JUMGRID code, it is

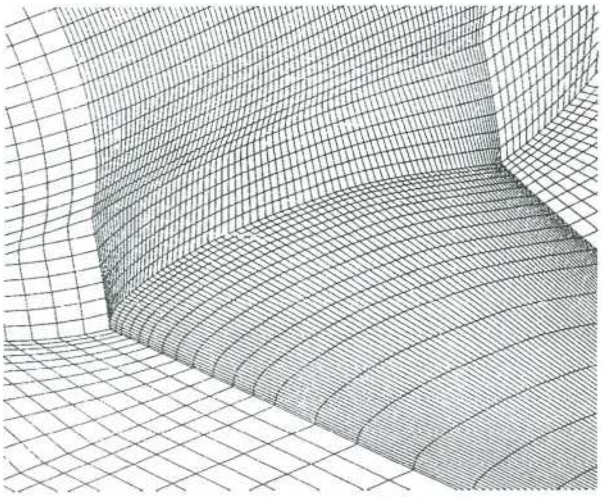


Fig. 9c Zoomed view of surface and field grid near wing fuselage junction SARAS wing fuselage stub-wing

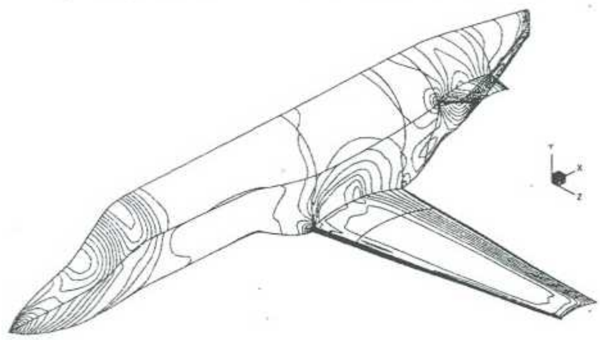


Fig. 9d Surface Mach contours for SARAS wing fuseInge stub-wing. M = 0.5, alpha = 0

necessary to use some other post-processing software for grid visualisation. Both of these codes are capable of producing grids of similar grid quality. Ultimately, it depends on how effectively the user has utilised the capabilities of the code used to generate the grid. There is no substantial difference in the quality of the solutions obtained on the grids generated by both JUMGRID and GRIDGEN.

Conclusions

Structured multi-block grid generation, although suitable for complex aerospace configurations, requires a lot of time and effort. Unstructured grid generation is much faster, and the effort required to develop 3D unstructured grid solvers will certainly be worthwhile. There are still some questions concerning the accuracy of viscous flow simulations using unstructured grids, and this is currently an active area of research. There is always the possibility of using hybrid grids in which a structured grid is inserted in the vicinity of the solid boundaries.

There is also remarkable progress in the use of adapted Cartesian grids for full aircraft configurations. Such grids usually use sophisticated flow solvers using the latest data structures and programming languages. Since the generation of Cartesian grids is extremely fast, such a grid generation cum solver package is also an extremely useful tool.

Overlapped or Chimera grids also simplify the grid generation process. However, the use of Overlapped rids requires a great deal of care in the interpolation procedure used to transfer data between grids, which is a source of inaccuracy. Even then, such grids have been used succes,-fully for extremely complex problems-

Ex isting finite volume flow solvers can he more easily modified to run on hybrid grids or adapted Cartesian grids than on overlapped grids-

Acknowledgements

The two-dimensional results shown in this paper form part of the author's Ph.D. research work under the guidance of Prof. S. M. Deshpande, Department of Aerospace Engineering, Indian Institute of Science (IISc), Bangalore, and Dr. S. S. Desai, Head. Computational and Theoretical Fluid Dynamics Division, National Aerospace Lahoratorics (NAL), Bangalore. All the three-dimensional grids shown have been developed by a team at NAL consisting of Dr. S. K. Chakrahartty and Mrs. K. Dhanalakshmi along with the author.

References

I. Weatherill, N. P., "Grid Generation", VKI Lecture Series 1990-10, June 1 I-15, 1990.

2. Thompson, J. F. and Weatherill, N. P., "Aspects of Numerical Grid Generation: Current Science and Art", AIAA Paper 93-3539, 1993.

3. Sickles, W. L. et al., "Separation Analysis of the Pegasus XL from an L-1011 Aircraft", Proceedings of 14th International Conference on Numerical Methods in Fluid Dynamio. Bangalore. India, July 11-15, 1994. Lecture Notes in Physics, Vol. 453, Springer, pp. 530-534, 1995,

4. Berger, M. and Aftosmis, M., "Aspects (and Aspect Ratios) of Cartesian Mesh Methods", Proceedings of

16th International Conference on Numerical Methods in Fluid Dynamics, Arcachon, France, July 6-10, 1998, Lecture Notes in Physics, Vol. 515, Springer, pp. 1-12, 1998 .

5. Mandal, J. C. and Deshpande, S. M., "Kinetic Flux Vector Splitting for Euler Equations", Computers and Fluids, Vol. 23, No. 2, pp. 447-478, 1994.

6. Mathur, J. S. and Weatherill, N. P., "The Simulation of Inviscid, Compressible Flow using an Upwind Kinetic Method on Unstructured Grids", International Journal for Numerical Methods in Fluids, Vol. 15, No. 1, pp. 59-82, July 1992.

7. Weatherill, N. P., Mathur, J. S. and Marchant, M. J., "An Upwind Kinetic Flux Vector Splitting Method on General Mesh Topologies", International Journal for Numerical Methods in Engineering, Vol. 37, No. 4, pp. 623-643, February 1994.

8. Mathur, J. S. and Deshpande, S. M., "Reconstruction on Unstructured Grids using an Upwind Kinetic Method", Proceedings of 15th International Conference on Numerical Methods in Fluid Dynamics, Monterey, California, June 24-28, 1996, Lecture Notes in Physics, Vol. 490, Springer, pp. 623-628, 1997.

9. Mathur, J. S. and Ramesh, V., "Computation of Inviscid Compressible Flow on Adapted Cartesian Grids using the Kinetic Flux Vector Splitting Method", Computational Fluid Dynamics Journal, Vol. 6, No. 1, pp. 51-64, April 1997 .

10. Mathur, J. S., "Application of Kinetic Schemes to Structured, Unstructured, Cartesian and Hybrid Grids", Ph.D. Thesis, Indian Institute of Science, Bangalore, November 1997.

11. Chakrabartty, S. K., Dhanalakshmi, K. and Mathur, J. S., "Computation of Three-dimensional Transonic Flow using a Cell Vertex Finite Volume Method for the Euler Equations", Acta Mechanica, Vol. 115, pp. 161-177, 1996.

12. Chakrabartty, S. K., Dhanalakshmi, K. and Mathur, J. S., "Computation of Three-dimensional Transonic Viscous Flow using the JUMBO3D Code", Acta Mechanica, Vol. 119, pp. 181-197, 1996.

13. Chakrabartty, S. K., Dhanalakshmi, K. and Mathur, J. S., "Navier-Stokes Analysis of Vortex Flow Over a Cropped Delta Wing", Acta Mechanica, Vol. 131, pp. 69-87, 1998.

14. Chakrabartty, S. K., Dhanalakshmi, K. and Mathur, J. S., "Euler Analysis of Multi-body Launch Vehicle Configurations", Proceedings of 3rd Asian Computational Fluid Dynamics Conference, Bangalore, Vol. 2, pp. 1-6, December 7-11, 1998.

15. Chakrabartty, S. K., Dhanalakshmi, K. and Mathur, J. S., "JUMGRID - a Multi-block Grid Generation Code", Proceedings of 3rd Asian Computational Fluid Dynamics Conference, Bangalore, Vol. 2, pp. 62-67, December 7-11, 1998.